Middleware Evaluation and Benchmarking for Mission Operations Centers

Ground System Architecture Workshop

March 1, 2005

Rob Antonucci
Emergent Space Technologies
RJA@emergentspace.com

Waka A. Waktola
NASA Goddard Space Flight Center
Waka.A.Waktola@nasa.gov

GSFC Mission Services Evolution Center
http://gmsec.gsfc.nasa.gov

Outline

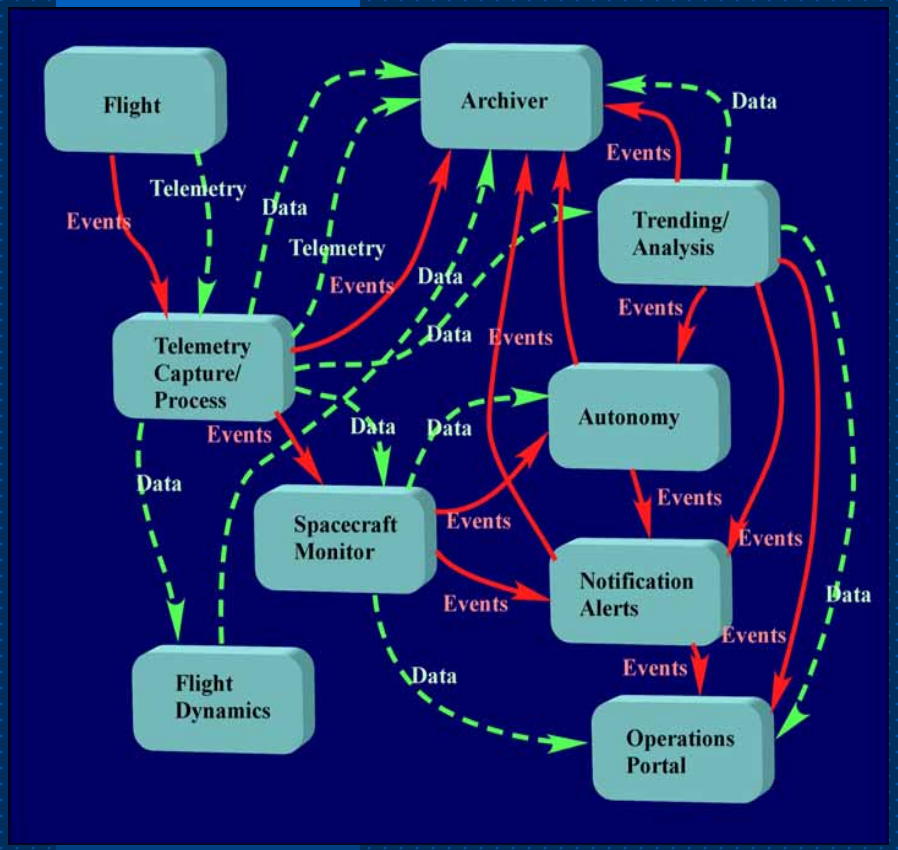
- **GMSEC Project and Middleware**
- **Middleware Performance Study**
  - Goals and Approach
  - Findings
  - Middleware Perceptions
- **Key Design Considerations**
- **Summary**

GSFC Mission Services Evolution Center (GMSEC)

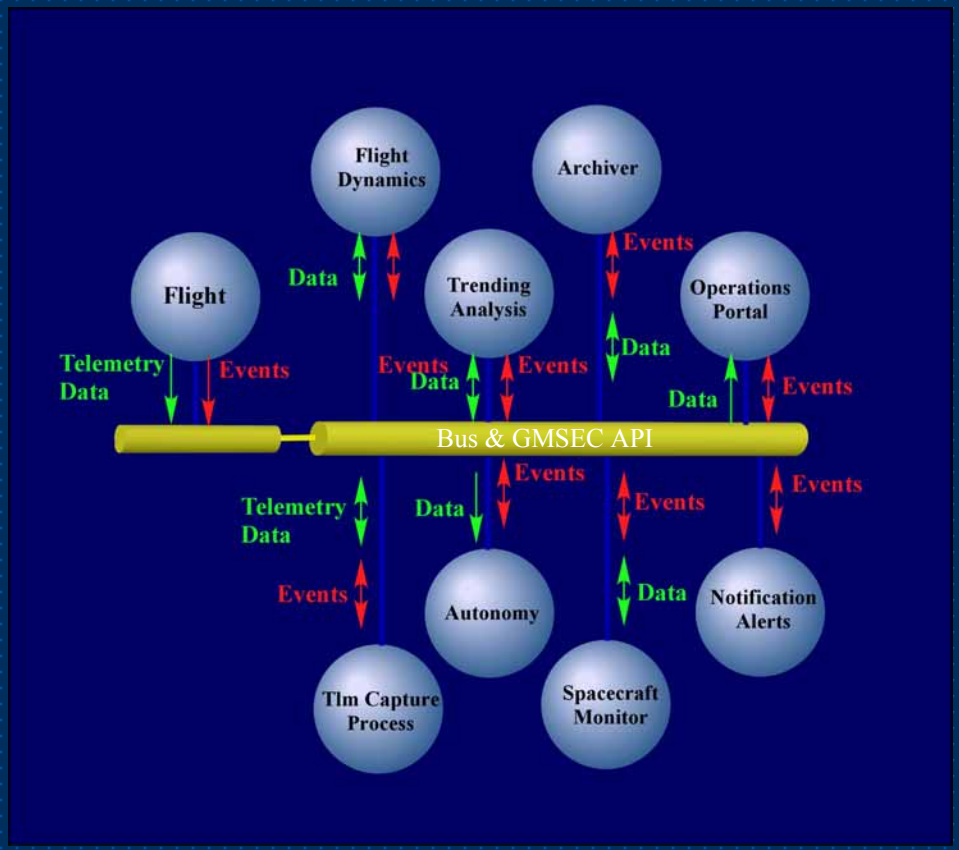
- **Next generation architecture to provide flexible and cost-effective mission services to meet GSFC mission needs**
  - Simplified integration of ground and flight software components
  - Support for evolving operational requirements
  - Simplified infusion of new technologies and components
- **Architecture must have core capability to add, swap and reconfigure individual software components without impact to remaining architecture**
- **Key strategy in meeting that capability is the reliance on middleware for communication and data requirements**

Middleware in GMSEC Domain

Socket Connections



Middleware Connections



Middleware Performance Study

- **Performance study started in 2004**
  - Evaluate and assess candidate middleware products
  - Compare/contrast middleware with point-to-point solutions
  - Validate/refute commonly held perceptions regarding viability of middleware solutions
- **Study performed in two phases**
  - Benchmarking to provide statistical metrics
  - Mission Operations Center (MOC) simulation to provide more realistic operational sanity check

Performance Study Approach

- **Benchmarking**
  - Tight monitoring of all data transmissions on a set of clients producing and consuming generic data across the middleware
- **MOC Simulation**
  - Replication of ground system environment with middleware delivering mission data
- **Target specific areas of assessment**
  - Delay and reliability
  - Impact of large messages
  - Validation with MOC simulation
- **Address middleware perceptions**
  - Overhead, Guaranteed Messages, Plug and Play, Cost

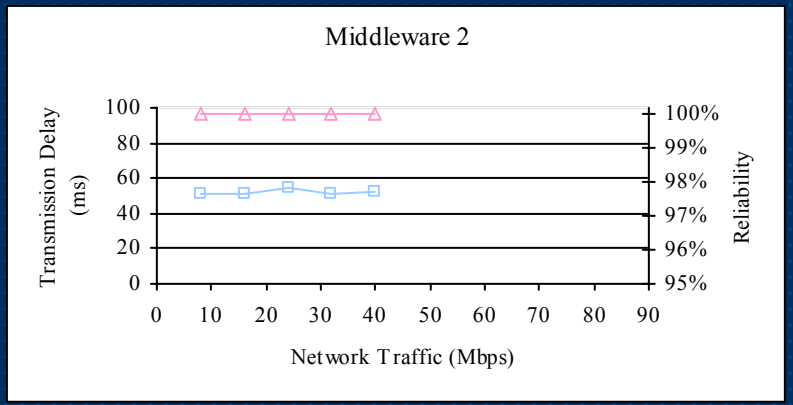
Candidate Middleware Products

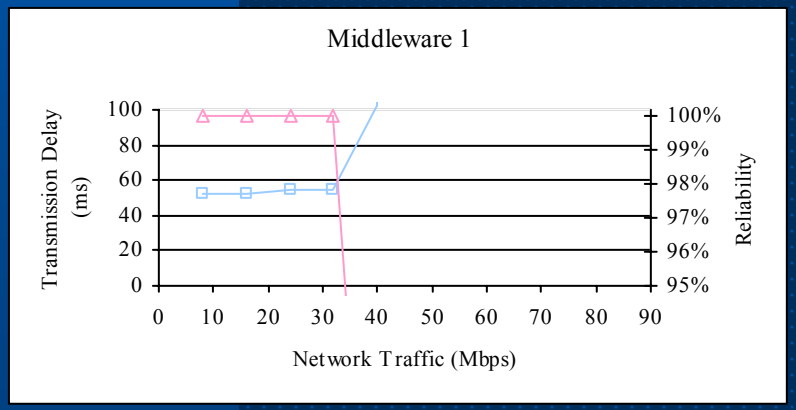
- **IBM Websphere***
- **ICS Software Bus**
- **Mantara Elvin***
- **TIBCO Smartsockets**
- **TIBCO Rendezvous**

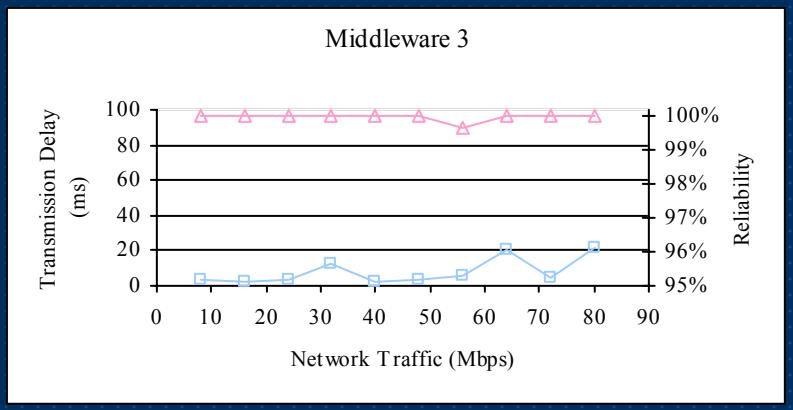
* Surveyed but not yet tested

Assessment of Delay and Reliability

- **Baseline Configuration**
  - 6 clients on 3 Windows 2000
  - 1 server on Windows 2000
- **Performance Requirements**
  - < 100 ms transmission delay
  - > 99.5% reliability
  - For loads 0-20 Mbps



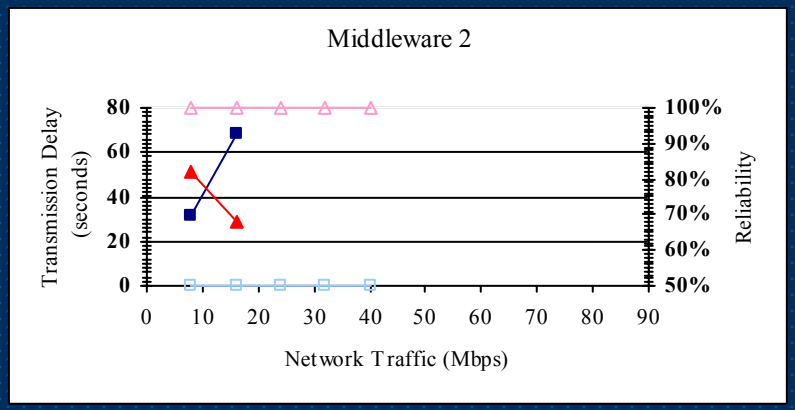


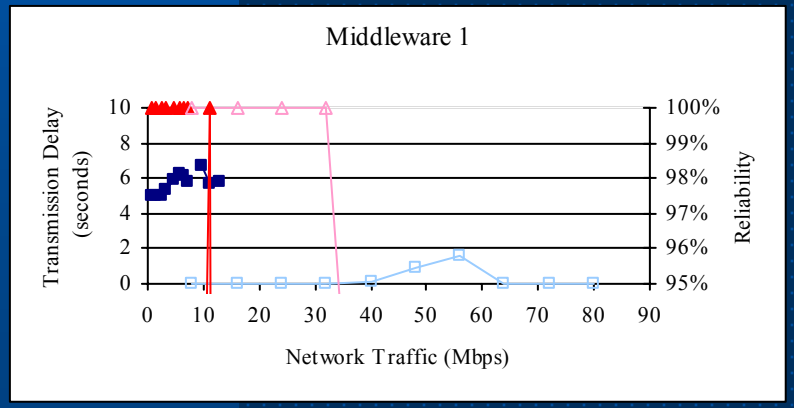


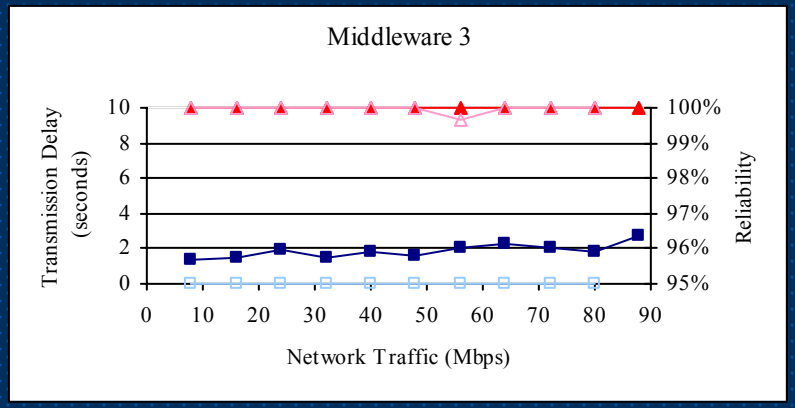
Transmission Delay Reliability

Assessment of Large Message Impact

- **5MB Message Configuration**
  - Use of multi-megabyte messages should be avoided if possible
  - If middleware does not support very large messages, packets can easily be broken into many smaller messages







Validation with MOC Simulation

- **Simulated GPM Mission Configuration**
  - T&C System
  - Event Analysis System
  - Simulated Trending/Archiving System
  - Operational TRMM telemetry data
  - Simulated TRMM science data producers and consumers
- **MOC simulation showed no errors or stress on system for tested middleware**

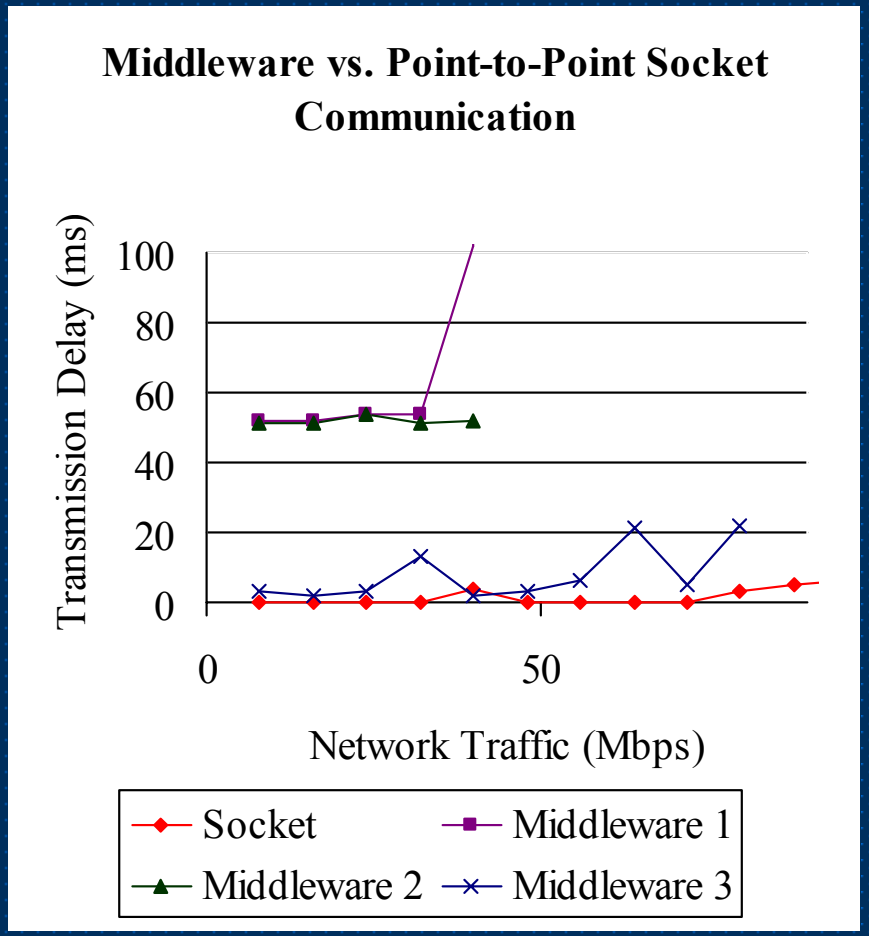
Middleware Perception: Overhead

User Perception:

- Middleware will impose significant time and throughput overhead

Reality:

- Time impact negligible and throughput still exceeds mission needs



Middleware Perception: Guaranteed Messages

User Perception:

- Guaranteed messages means all messages will be successful

Reality:

- Client will be informed if message is not successful
- Extra effort can ensure that message is delivered
  - Point-to-point confirmation for regular messages
  - End-to-end confirmation added for guaranteed messages
  - Clients may have their own end-to-end confirmation mechanism (request/response)
- Messages cached to disk will survive crash

Middleware Perception: Plug and Play

User Perception:

- Middleware is instant interoperability

Reality:

- Connection to middleware requires component modifications
- Bridging applications can limit scope of changes
- GMSEC API standardizes interface and behavior to middleware and data model common to all components

Middleware Perception: Cost

User Perception:

- Middleware solutions make architecture cost-prohibitive

Reality:

- There is wide cost variation among middleware products
- Required capabilities may need to be closely examined to find best fit

	SmartSockets	Rendezvous	WebSphere	Elvin	ICS
Fault Tolerance	Server + Client	Server + Client	Server + Client	Server	No
Load Balancing	Server + Client	Server + Client	Server + Client	Server	No
Guaranteed Messages	Yes	Yes	Yes	Yes	No
Cost	$$$	$$$	$$	$	$

Key Design Considerations

When Should Middleware Be Used?

Pros

Easy to add or swap out components

Less integration time

Best For

New missions

Long lived missions

Low budget missions

Missions with changing requirements

Cons

Existing components must migrate

May require development

COTS middleware mandate upgrades

Worst For

Existing missions with short life expectancy due to re-engineering costs

Key Design Considerations

Should Messages Be Guaranteed?

Pros

More reliable

Removes single point of failure

Sender can react if never received

Cons

Poorer performance

May be repeating effort of client

Due to timeliness, may not want messages to survive crash

Best For

Critical messages

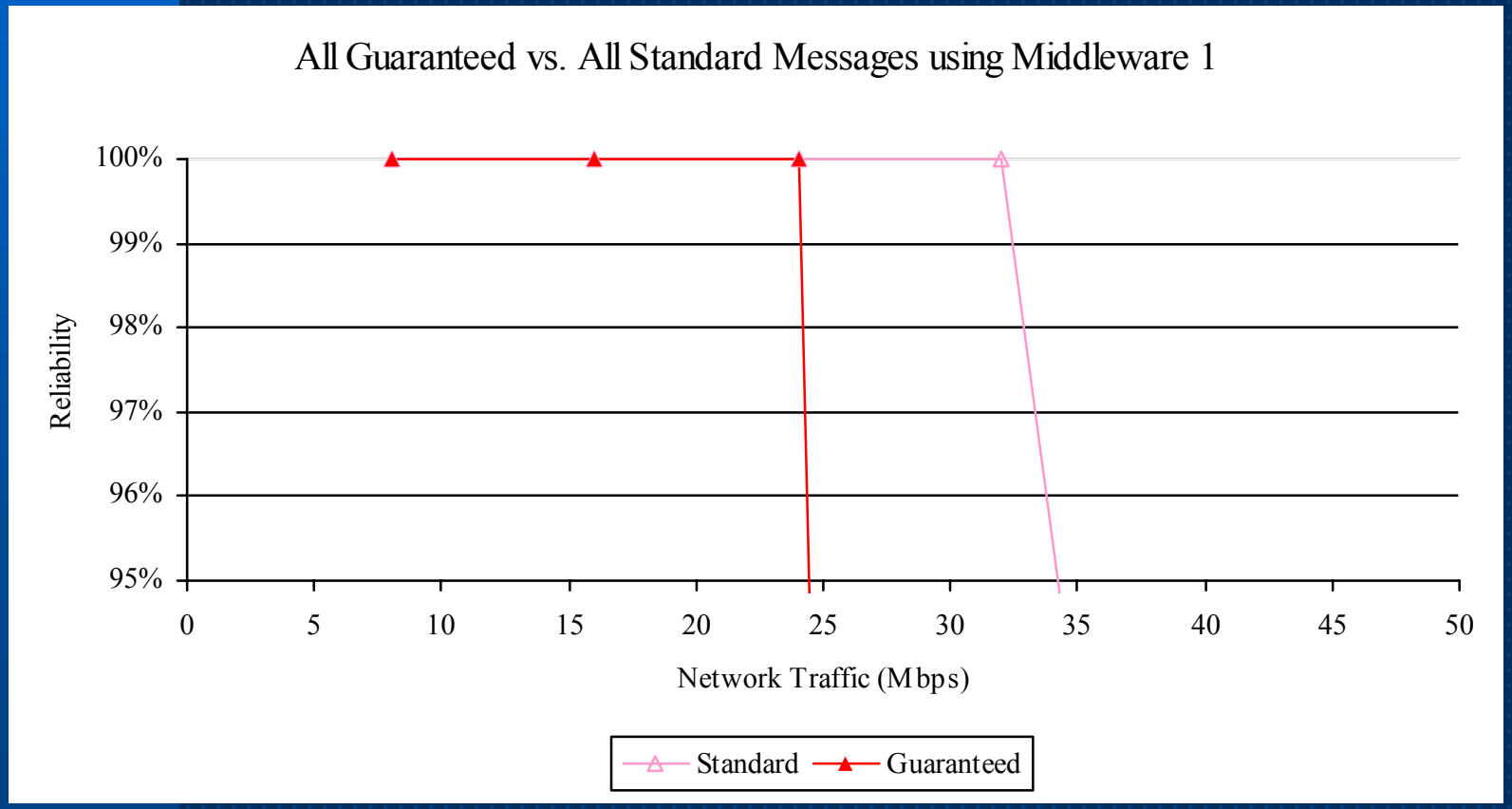
Messages that sender can react to if never received

Worst For

Time sensitive information

High frequency information

Impact of Guaranteed Messages



Too many guaranteed messages actually reduce overall success rate.

Key Design Considerations

What Other Characteristics Should Be Considered?

- **Should servers be redundant?**
  - Redundancy not offered in less expensive products
  - Best used for autonomy that cannot support a single point of failure
- **Should ground systems use middleware redundancy?**
  - Component redundancy only offered in more expensive products
  - Best used for critical components
- **What if the expected load exceeds benchmark limits?**
  - Some middleware supports load balancing
  - Multiple servers splitting load

Summary

- Middlewares are capable of performing in a mission operational environment
- Cost-effective middleware solutions available for all types of missions
- Middleware-based architectures are flexible to support evolving mission requirements

Acronyms

API	Applications Programming Interface
COTS	Commercial-Off-The-Shelf
GMSEC	GSFC Mission Services Evolution Center
GPM	Global Precipitation Measurement
GSFC	Goddard Space Flight Center
ICS	Interface & Control Systems, Inc.
T&C	Telemetry and Command
TRMM	Tropical Rainforest Measurement Mission